Claim Amendments

Claim 1 (currently amended): A data storage system comprising:

N servers, where N $\geq$ 2 and is an integer, each server includes a state machine and a memory, and maintains in the memory a last time at which each servers' state changed and a value associated with the state when it last changed;

D disks, where D $\geq$ 2 and is an integer, each server in communication with each disk, each disk having a reserved disk block for each of the N servers; and

a disk arbitration mechanism that uses a timestamp-based voting algorithm over the disk blocks associated with the servers to exchange votes for a primary server, the primary server arbitrating access of the servers to a set of disks of the D disks, where a set of the disk blocks are used as a communication medium.

Claim 2 (canceled)

Claim 3 (previously presented): A system as described in Claim 1 wherein each server has an index.

Claim 4 (original): A system as described in Claim 3 wherein the disk arbitration mechanism causes each server at first predetermined times to read all of the disk blocks, and write its own disk block to determine which server has access to, and use and control of the disks at a given time.

Claim 5 (original): A system as described in Claim 4 wherein the memory is ~~each server includes a state machine and~~ a local RAM, and each server maintains in local RAM **[[a]]** the last time at which each servers' state changed and **[[a]]** the value associated with the state when it last changed.

Claim 6 (original): A system as described in Claim 5 wherein each server determines which of the other servers are operating by identifying which of the other servers had their state change during second predetermined times.

Claim 7 (currently amended): A method for storing data comprising the steps of:

writing by N **[[of]]** servers into each servers' own associated disk block in each disk of D disks its state, a last time at which each servers' state changed and a value associated with the state when it last changed, where $N \geq 2$ and $D \geq 2$ and are integers; and

reading by each server all the other servers' disk blocks in each disk in order to determine which server has access to, and use and control of the disks at a given time by using a timestamp-based voting algorithm over the disk blocks associated with the servers and the servers' state to exchange votes for a primary server, the primary server arbitrating access of the servers to a set of disks of the D disks, where the set of the disk blocks are used as a communication medium.

Claim 8 (original): A method as described in Claim 7 wherein the reading step includes the step of performing a voting protocol to determine which server has access to, and use and control of the disks at a given time.

Claim 9 (original): A method as described in Claim 8 including after the reading step, there are the steps of determining which server becomes a winning server and has access to, and use and control of the disk at a given time; and accessing the disk exclusively by the winning server.

Claim 10 (original): A method as described in Claim 9 wherein the accessing step includes the step of transmitting by the winning server its state from not winning to winning and invalidating by the winning server all caches of the winning server.

Claim 11 (original): A method as described in Claim 10 wherein the writing step includes the step of assigning an index to each server.

Claim 12 (original): A method as described in Claim 11 wherein the reading step includes the step of reading at predetermined times by each server all disk blocks and writing its own respective disk block.

Claim 13 (original): A method as described in Claim 12 wherein the writing step includes the step of maintaining by each server in each servers' own local RAM a last time for each other server when each other servers' status changed and a value of a status counter at the last time.

Claim 14 (original): A method as described in Claim 13 wherein the reading step includes the step of determining by each server which of the other servers are operating by declaring that each of the other servers whose status has changed within a last predetermined time period is operating.

Claim 15 (original): A method as described in Claim 14 wherein the reading step includes the step of voting by the servers that are up for a winning server that is up and believes it is the winning server.

Claim 16 (original): A method as described in Claim 15 wherein the reading step includes the step of voting for the server that is up and has a lowest index if no server believes it is the winning server.

Claim 17 (previously presented): A method for storing data comprising the steps of:

writing by N of servers into each servers' own associated disk block in each disk of D disks its state, where $N \geq 2$ and $D \geq 2$ and are integers; and

reading by each server all the other servers' disk blocks in each disk in order to determine which server has access to, and use and control of the disks at a given time;

determining which server becomes a winning server and has access to, and use and control of the disk at a given time;

accessing the disk exclusively by the winning server; and

transmitting by the winning server its state from not winning to winning and invalidating by the winning server all caches of the winning server.

Claim 18 (previously presented): A method as described in Claim 17 wherein the writing step includes the step of assigning an index to each server.

Claim 19 (previously presented): A method as described in Claim 18 wherein the reading step includes the step of reading at predetermined times by each server all disk blocks and writing its own respective disk block.

Claim 20 (previously presented): A method as described in Claim 19 wherein the writing step includes the step of maintaining by each server in each servers' own local RAM a last time for each other server when each other servers' status changed and a value of a status counter at the last time.

Claim 21 (previously presented): A method as described in Claim 20 wherein the reading step includes the step of determining by each server which of the other servers are operating by declaring that each of the other servers whose status has changed within a last predetermined time period is operating.

Claim 22 (previously presented): A method as described in Claim 21 wherein the reading step includes the step of voting by the servers that are up for a winning server that is up and believes it is the winning server.

Claim 23 (previously presented): A method as described in Claim 22 wherein the reading step includes the step of voting for the server that is up and has a lowest index if no server believes it is the winning server.

Claim 24 (previously presented): A data storage system comprising:

N servers, where $N \geq 2$ and is an integer, each server has an index, each server includes a state machine and a local RAM, and maintains in local RAM a last time at which each servers' state changed and a value associated with the state when it last changed;

D disks, where $D \geq 2$ and is an integer, each server in communication with each disk, each disk having a reserved disk block for each of the N servers; and

a disk arbitration mechanism where each of the N servers writes its state in its own associated disk block in each disk, and reads all the other servers' disk blocks in each disk in order to determine which server has access to, and use and control of the disks at a given time, the disk arbitration mechanism causes each server at first predetermined times to read all of the disk blocks, and write its own disk block to determine which server has access to, and use and control of the disks at a given time.

Claim 25 (previously presented): A system as described in Claim 24 wherein each server determines which of the other servers are operating by identifying which of the other servers had their state change during second predetermined times.

Claim 26 (new): A system as described in Claim 6 wherein in the state machine, a transition from state 1 to state 2 occurs if the server has been in state 1 for a minimum of tMax seconds, and during that time, has seen itself as the server that is both functioning and has the lowest index of the functioning servers.

Claim 27 (new): A system as described in Claim 26 wherein in the state machine, a transition from state 2 to state 3 occurs after the server sees that it has the votes of a proper majority of the servers, including itself.

Claim 28 (new): A system as described in Claim 27 wherein in the state machine, a transition from state 1 to state 4 occurs after a minimum of tMax seconds if the server sees that another server is the server that is both functioning and has the lowest index of all functioning servers.

Claim 29 (new): A system as described in Claim 28 wherein in the state machine, a transition from state 4 to state 5 occurs if the server sees that the other server has declared itself the winner of the election.

Claim 30 (new): A system as described in Claim 29 wherein in the state machine, a transition directly from state 1 to state 5 occurs if a minimum of tMax has passed and during that time, the other server has declared that it has won the election; a transition from state 5 to state 1 occurs if the server that has declared it has won the election is no longer functioning, or no longer declares that it has won the election; a transition from state 4 to state 1 occurs if the other server no longer appears functioning; a transition from state 2 to state 1 occurs if the server has not yet won the election, and the other server appears with a lower index than the server; a transition from state 3 to state 1 occurs if the server sees the other server with a lower index decide that the other server has won the election.

Claim 31 (new): A method as described in Claim 16 including the step of transitioning in the state machine from state 1 to state 2 if the server has been in state 1 for a minimum of tMax seconds, and during that time, has seen itself as the server that is both functioning and has the lowest index.

Claim 32 (new): A method as described in Claim 31 including the step of transitioning in the state machine from state 2 to state 3 after a server sees that it has the votes of a proper majority of the servers, including itself.

Claim 33 (new): A method as described in Claim 32 including the step of transitioning in the state machine from state 1 to state 4 after a minimum of tMax seconds if the server sees that another server is the server that is both functioning and has the lowest index of all functioning servers.

Claim 34 (new): A method as described in Claim 33 including the step of transitioning in the state machine from state 4 to state 5 occurs if the server sees that the other server has declared itself the winner of the election.

Claim 35 (new): A method as described in Claim 34 including the step of transitioning in the state machine directly from state 1 to state 5 if a minimum of tMax has passed and during that time, the other server has declared that it has won the election; from state 5 to state 1 occurs if the server that has declared it has won the election no longer appears functioning, or no longer declares that it has won the election; from state 4 to state 1 if the server no longer appears functioning; from state 2 to state 1 occurs if the server has not yet won the election, and the other server appears with a lower index that the server; from

state 3 to state 1 occurs if it sees the other server with a lower index decide that it has won the election.